GHKM Rule Extraction and Scope-3 Parsing in Moses

GHKM Rule Extraction and Scope-3 Parsing in Moses

Philip Williams and **Philipp Koehn**
School of Informatics
University of Edinburgh
10 Crichton Street
EH8 9AB, UK
p.j.williams-2@sms.ed.ac.uk
pkoehn@inf.ed.ac.uk

Abstract

We developed a string-to-tree system for English–German, achieving competitive results against a hierarchical model baseline. We provide details of our implementation of GHKM rule extraction and scope-3 parsing in the Moses toolkit. We compare systems trained on the same data using different grammar extraction methods.

1 Introduction

Over the last few years, syntax-based rule extraction has largely developed along two lines, one originating in hierarchical phrase-based translation (Chiang, 2005; Chiang, 2007) and the other in GHKM (Galley et al., 2004; Galley et al., 2006).

Hierarchical rule extraction generalizes the established phrase-based extraction method to produce formally-syntactic synchronous context-free grammar rules without any requirement for linguistic annotation of the training data. In subsequent work, the approach has been extended to incorporate linguistic annotation on the target side (as in SAMT (Zollmann and Venugopal, 2006)) or on both sides (Chiang, 2010).

In contrast, GHKM places target-side syntactic structure at the heart of the rule extraction process, producing extended tree transducer rules that map between strings and tree fragments.

Ultimately, both methods define rules according to a sentence pair's word-alignments. Without any restriction on rule size they will produce an exponentially large set of rules and so in practice only a subgrammar can be extracted. It is the differing rule selection heuristics that distinguish these two approaches, with hierarchical approaches being motivated by phrasal coverage and GHKM by target-side tree coverage.

The Moses toolkit (Koehn et al., 2007) has included support for hierarchical phrase-based rule extraction since the decoder was first extended to support syntax-based translation (Hoang et al., 2009). In this paper we provide some implementation details for the recently-added GHKM rule extractor and for the related scope-3 decoding algorithm. We then describe the University of Edinburgh's GHKM-based English-German submission to the WMT translation task and present comparisons with hierarchical systems trained on the same data. To our knowledge, these are the first GHKM results presented for English-German, a language pair with a high degree of reordering and rich target-side morphology.

2 GHKM Rule Extraction in Moses

A basic GHKM rule extractor was first developed for Moses during the fourth Machine Translation Marathon[1] in 2010. We have recently extended it to support several key features that are described in the literature, namely: composition of rules (Galley et al., 2006), attachment of unaligned source words (Galley et al., 2004), and elimination of fully non-lexical unary rules (Chung et al., 2011).

We provide some basic implementation details in the remainder of this section. In section 4 we present

[1] http://www.mtmarathon2010.info

Proceedings of the 7th Workshop on Statistical Machine Translation, pages 388–394,
Montréal, Canada, June 7-8, 2012.

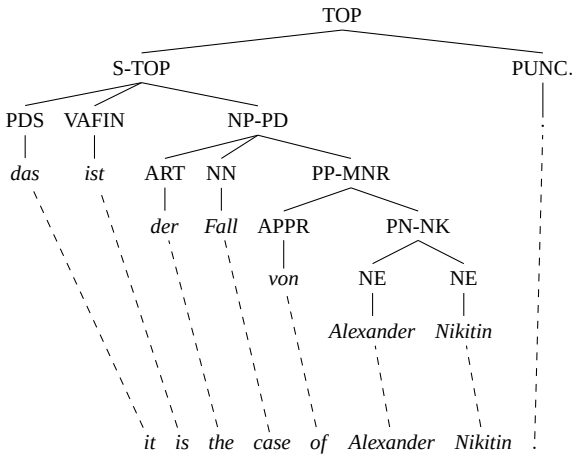

Figure 1: Sentence pair from training data.

experimental results comparing performance against Moses' alternative rule extraction methods.

2.1 Composed Rules

Composition of minimal GHKM rules into larger, contextually-richer rules has been found to significantly improve translation quality (Galley et al., 2006). Allowing any combination of adjacent minimal rules without restriction is unfeasible and so in practice various constraints are imposed on composition. Our implementation includes three configurable parameters for this purpose, which we describe with reference to the example alignment graph shown in Figure 1. All three are defined in terms of the target tree fragment.

Rule depth is defined as the maximum distance from the composed rule's root node to any other node within the fragment, not counting preterminal expansions (such as NE → *Nikitin*). By default, the rule depth is limited to three. If we consider the composition of rules rooted at the S-TOP node in Figure 1 then, among many other possibilities, this setting permits the formation of a rule with the target side:

S-TOP → *das ist der Fall von* PN-NK

since the maximum distance from the rule's root node to another node is three (to APPR or to PN-NK). However, a rule with the target side:

S-TOP → *das ist der Fall von* NE *Nikitin*

is not permitted since it has a rule depth of four (from S-TOP to either of the NE nodes).

Node count is defined as the number of target tree nodes in the composed rule, excluding target words. The default limit is 15, which for the example is large enough to permit any possible composed rule (the full tree has a node count of 13).

Rule size is the measure defined in DeNeefe et al. (2007): the number of non-part-of-speech, non-leaf constituent labels in the target tree. The default rule size limit is three.

2.2 Unaligned Source Words

Unaligned source words are attached to the tree using the following heuristic: if there are aligned source words to both the left and the right of an unaligned source word then it is attached to the lowest common ancestor of its nearest such left and right neighbours. Otherwise, it is attached to the root of the parse tree.

2.3 Unary Rule Elimination

Moses' chart decoder does not currently support the use of grammars containing fully non-lexical unary rules (such as NP → $X_1 \mid NN_1$). Unless the `--AllowUnary` option is given, the rule extractor eliminates these rules using the method described in Chung et al. (2011).

2.4 Scope Pruning

Unlike hierarchical phrase-based rule extraction, GHKM places no restriction on the rank of the resulting rules. In order that the grammar can be parsed efficiently, one of two approaches is usually taken: (i) *synchronous binarization* (Zhang et al., 2006), which transforms the original grammar to a weakly equivalent form in which no rule has rank greater than two. This makes the grammar amenable to decoding with a standard chart-parsing algorithm such as CYK, and (ii) *scope pruning* (Hopkins and Langmead, 2010), which eliminates rules in order to produce a subgrammar that can be parsed in cubic time.

Of these two approaches, Moses currently supports only the latter. Both rule extractors prune the extracted grammar to remove rules with scope greater than three. The next section describes the parsing algorithm that is used for scope-3 grammars.

3 Scope-3 Parsing in Moses

Hopkins and Langmead (2010) show that a sentence of length n can be parsed using a scope-k grammar in $O(n^k)$ chart updates. In this section, we describe some details of Moses' implementation of their chart parsing method.

3.1 The Grammar Trie

The grammar is stored in a trie-based data structure. Each edge is labelled with either a symbol from the source terminal vocabulary or a generic gap symbol, and the trie is constructed such that for any path originating at the root vertex, the sequence of edge labels represents the prefix of a rule's source right-hand-side (RHS_s, also referred to as a rule pattern). Wherever a path corresponds to a complete RHS_s, the vertex stores an associative array holding the set of grammar rules that share that RHS_s. The associative array maps a rule's sequence of target non-terminal symbols to the subset of grammar rules that share those symbols.

Figure 2 shows a sample of the grammar rules that can be extracted from the example alignment graph of Figure 1, and Figure 3 shows the corresponding grammar trie.

3.2 Initialization

The first step is to construct a secondary trie that records all possible applications of rule patterns from the grammar to the sentence under consideration. This trie is built during a single depth-first traversal of the grammar trie in which the terminal edge labels are searched for in the input sentence. If a matching input word is found then the secondary trie is extended by one vertex for each sentence position at which the word occurs and trie traversal continues along that path. A search for a gap label always results in a match. Edges in the secondary trie are labelled with the matching symbol and the position of the word in the input sentence (or a null position for gap labels). Each vertex in the secondary trie stores a pointer to the corresponding grammar trie vertex.

Once the secondary trie has been built, it is easy to determine the set of subspans to which each rule pattern applies. A set of pairs is recorded against each subspan, each pair holding a pointer to a grammar trie vertex and a record of the sentence positions covered by the symbols (which will be ambiguous if the pattern contains a sequence of $k > 1$ adjacent gap symbols covering more than k sentence positions).

After this initialization step, the secondary trie is discarded.

3.3 Subspan Processing

The parsing algorithm proceeds by processing chart cells in order of increasing span width (i.e. bottom-up). At each cell, a *stack lattice* is constructed for each rule pattern that was found during initialization. The stack lattice compactly represents all possible applications of that pattern over the span, together with pointers to the underlying hypothesis stacks for every gap. A full path through the lattice corresponds to a single application context. By selecting a derivation class (i.e. target-side non-terminal label) at each arc, the path can be bound to a set of grammar rules that differ only in the choice of target words or LHS label.

Recall that for every rule pattern found during initialization, the corresponding grammar trie vertex was recorded and that the vertex holds an associative array in which the keys are sequences of target-side non-terminal labels and the mapped values are grammar rules (together with associated feature model scores). The algorithm now loops over the associated array's key sequences, searching the lattice for matching paths. Where found, the grammar rule is bound with a sequence of underlying stack pointers. The cell's stacks are then populated by applying cube pruning (Chiang, 2007) to the set of bound grammar rules.

4 Experiments

This section describes the GHKM-based English-German system submitted by the University of Edinburgh. Subsequent to submission, a further set of comparative experiments were run using a hierarchical phrase-based system and a hierarchical system with target side syntactic annotation.

4.1 Data

We made use of all available English-German European and News Commentary data. For the hierarchical phrase-based experiments, this totalled

1. NP-PD → *the case of Alexander Nikitin* | *der Fall von Alexander Nikitin*
2. NP-PD → *the case* X_1 | *der Fall* PP-MNR$_1$
3. NP-PD → X_1 *case* X_2 | ART$_1$ *Fall* PP-MNR$_2$
4. PP-MNR → *of* X_1 | *von* PN-NK$_1$
5. PP-MNR → *of* X_1 X_2 | *von* NE$_1$ NE$_2$

Figure 2: A sample of the rules extractable from the alignment graph in Figure 1. Rules are written in the form LHS → RHS$_s$ | RHS$_t$.

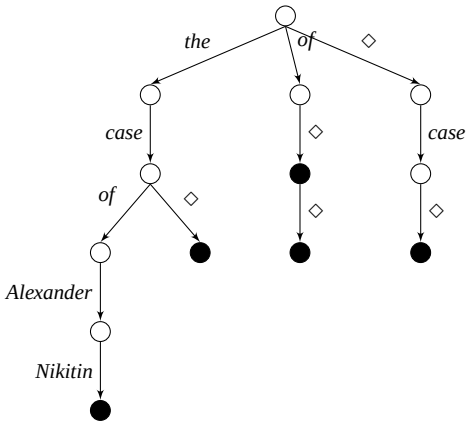

Figure 3: Example grammar trie. The filled vertices hold associative array values.

2,043,914 sentence pairs. For the target syntax experiments, the German-side of the parallel corpus was parsed using the BitPar[2] parser. If a parse failed then the sentence pair was discarded, leaving a total of 2,028,556 pairs. The parallel corpus was then word-aligned using MGIZA++ (Gao and Vogel, 2008), a multi-threaded implementation of GIZA++ (Och and Ney, 2003).

We used all available monolingual German data to train seven 5-gram language models (one each for Europarl, News Commentary, and the five News data sets). These were interpolated using weights optimised against the development set and the resulting language model was used in experiments. We used the SRILM toolkit (Stolcke, 2002) with Kneser-Ney smoothing (Chen and Goodman, 1998).

The baseline system's feature weights were tuned on the *news-test2008* dev set (2,051 sentence pairs) using Moses' implementation of minimum error rate training (Och, 2003).

[2] `http://www.ims.uni-stuttgart.de/tcl/SOFTWARE/BitPar.html`

4.2 Rule Extraction

For the hierarchical phrase-based model we used the default Moses rule extraction settings, which are taken from Chiang (2007). For target-annotated models, the syntactic constraints imposed by the parse trees reduce the grammar size significantly. This allows us to relax the rule extraction settings, which we have previously found to benefit translation quality, without producing an unusably large grammar. We use identical settings to those used in WMT's 2010 translation task (Koehn et al., 2010). Specifically, we relax the hierarchical phrase-based extraction settings in the following ways:

- Up to seven source-side symbols are allowed.
- Consecutive source non-terminals are permitted.
- Single-word lexical phrases are allowed for hierarchical subphrase subtraction.
- Initial phrases are limited to 15 source words (instead of 10).

By using the scope-3 parser we can also relax the restriction on grammar rank. For comparison, we extract two target-annotated grammars, one with a maximum rank of two, and one with an unlimited rank but subject to scope-3 pruning.

GHKM rule extraction uses the default settings[3] as described in section 2.

Table 1 shows the sizes of the extracted grammars after filtering for the `newstest2011` test set. Filtering removes any rule in which the source right-hand-side contains a sequence of terminals and gaps that does not appear in any test set sentence.

[3] GHKM rule extraction is now fully integrated into Moses' Experiment Management System (EMS) and can be enabled for string-to-tree pipelines using the `TRAINING:use-ghkm` parameter.

Experiment	Grammar Size
Hierarchical	118,649,771
Target Syntax	12,748,259
Target Syntax (scope-3)	40,661,639
GHKM	27,002,733

Table 1: Grammar sizes (distinct rule counts) after filtering for the `newstest-2011` test set

4.3 Features

Our feature functions include the n-gram language model probability of the derivation's target yield, its word count, and various scores for the synchronous derivation. We score grammar rules according to the following functions:

- $p(\mathrm{RHS}_s|\mathrm{RHS}_t, \mathrm{LHS})$, the noisy-channel translation probability.
- $p(\mathrm{LHS}, \mathrm{RHS}_t|\mathrm{RHS}_s)$, the direct translation probability.
- $p_{lex}(\mathrm{RHS}_t|\mathrm{RHS}_s)$ and $p_{lex}(\mathrm{RHS}_s|\mathrm{RHS}_t)$, the direct and indirect lexical weights (Koehn et al., 2003).
- $p_{pcfg}(\mathrm{FRAG}_t)$, the monolingual PCFG probability of the tree fragment from which the rule was extracted (GHKM and target-annotated systems only). This is defined as $\prod_{i=1}^{n} p(r_i)$, where $r_1 \ldots r_n$ are the constituent CFG rules of the fragment. The PCFG parameters are estimated from the parse of the target-side training data. All lexical CFG rules are given the probability 1. This is similar to the p_{cfg} feature used in Marcu et al. (2006) and is intended to encourage the production of syntactically well-formed derivations.
- $exp(-1/count(r))$, a rule rareness penalty.
- $exp(1)$, a rule penalty. The main grammar and glue grammars have distinct penalty features.

4.4 Decoder Settings

For the submitted GHKM system we used a maximum chart span setting of 25. For the other systems we used settings that matched the rule extraction spans: 10 for hierarchical phrase-based, 15 for target syntax, and unlimited for GHKM.

We used the scope-3 parsing algorithm (enabled using the option `-parsing-algorithm 1`) for all systems except the hierarchical system, which used the CYK+ algorithm (Chappelier and Rajman, 1998).

For all systems we set the `ttable-limit` parameter to 50 (increased from the default value of 20). This setting controls the level of grammar pruning that is performed after loading: only the top scoring translations are retained for a given source RHS.

4.5 Results

Following the recommendation of Clark et al. (2011), we ran the optimization three times and repeated evaluation with each set of feature weights. Table 2 presents the averaged single-reference BLEU scores. To give a rough indication of how much use the systems make of syntactic information for reordering, we also report glue rule statistics taken from the 1-best derivations.

There is a huge variation in decoding time between the systems, much of which can be attributed to the differing chart span limits. To give a comparison of system performance we selected an 80-sentence subset of `newstest2011`, randomly choosing ten sentences of length 1-10, ten of length 11-20, and so on. We decoded the test set four times for each system, discarding the first set of results (to allow for filesystem cache priming) and then averaging the remaining three. Table 3 shows the total decoding times for each system and the peak virtual memory usage[4]. Figure 4 shows a plot of sentence length against decoding time for the two GHKM systems.

5 Conclusion

We developed a GHKM-based string-to-tree system for English to German, achieving competitive results compared to a hierarchical model baseline. We extended the Moses toolkit to include a GHKM rule extractor and scope-3 parsing algorithm and provided details of our implementation. We intend to further improve this system in future work.

[4]The server has 142GB physical memory. The decoder was run single-threaded in performance tests. For the hierarchical system we used an on-disk rule table, which reduces memory requirements at the cost of increased rule lookup time. For all other systems we used in-memory rule tables.

Experiment	newstest2009 BLEU	s.d.	newstest2010 BLEU	s.d.	newstest2011 BLEU	s.d.	Glue Rule Apps Mean	s.d.
GHKM (max span 25)	**15.2**	0.1	**16.7**	0.1	15.4	0.1	3.1	0.3
Hierarchical	**15.2**	0.0	16.4	0.1	**15.5**	0.0	13.9	0.5
Target	14.6	0.1	16.0	0.1	14.9	0.1	8.4	5.0
Target (scope-3)	14.7	0.0	16.4	0.2	15.0	0.0	9.7	1.2
GHKM (no span limit)	15.0	0.3	16.6	0.1	15.2	0.2	1.9	1.3

Table 2: Average BLEU scores and standard deviations over three optimization runs. GHKM (max span 25) is the submitted system. Also shown is the average number of rule applications per sentence for the 1-best output of the three test sets, averaged over the three optimization runs.

System	Max span	Time (s)	VM (MB)
Hierarchical	10	122	5,345
Target	15	367	8,688
Target (scope-3)	15	1,539	19,761
GHKM	25	3,529	17,424
GHKM	None	11,196	18,060

Table 3: Total decoding time and peak virtual memory usage for the 80-sentence subset of `newstest2011`.

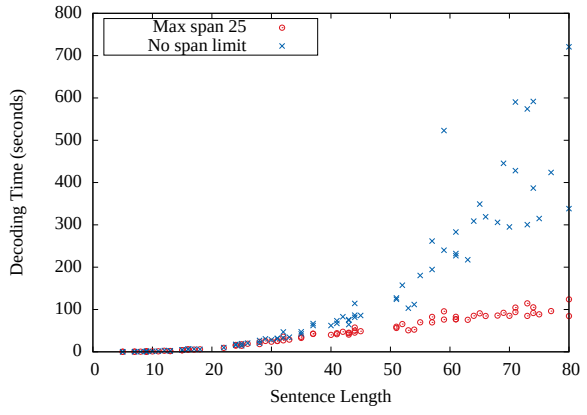

Figure 4: Sentence length vs decoding time for the GHKM (max span 25) and GHKM (no limit) systems

Acknowledgments

We would like to thank the anonymous reviewers for their helpful feedback and suggestions. This work was supported by the EuroMatrixPlus project funded by the European Commission (7th Framework Programme) and made use of the resources provided by the Edinburgh Compute and Data Facility.[5] The ECDF is partially supported by the eDIKT initiative.[6] This work was also supported in part under the GALE program of the Defense Advanced Research Projects Agency, Contract No. HR0011-06-C-0022. The first author was supported by an EPSRC Studentship.

References

J.-C. Chappelier and M. Rajman. 1998. A generalized CYK algorithm for parsing stochastic CFG. In *Proceedings of the First Workshop on Tabulation in Parsing and Deduction*, pages 133–137.

Stanley F. Chen and Joshua Goodman. 1998. An empirical study of smoothing techniques for language modeling. Technical report, Harvard University.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 263–270, Morristown, NJ, USA. Association for Computational Linguistics.

David Chiang. 2007. Hierarchical phrase-based translation. *Comput. Linguist.*, 33(2):201–228.

David Chiang. 2010. Learning to translate with source and target syntax. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguis-*

[5] `http://www.ecdf.ed.ac.uk`
[6] `http://www.edikt.org.uk`

tics, pages 1443–1452, Uppsala, Sweden, July. Association for Computational Linguistics.

Tagyoung Chung, Licheng Fang, and Daniel Gildea. 2011. Issues concerning decoding with synchronous context-free grammar. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 413–417, Portland, Oregon, USA, June. Association for Computational Linguistics.

Jonathan H. Clark, Chris Dyer, Alon Lavie, and Noah A. Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 176–181, Portland, Oregon, USA, June. Association for Computational Linguistics.

Steve DeNeefe, Kevin Knight, Wei Wang, and Daniel Marcu. 2007. What can syntax-based MT learn from phrase-based MT? In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007). June 28-30, 2007. Prague, Czech Republic.*

Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. 2004. What's in a translation rule? In *HLT-NAACL '04*.

Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. 2006. Scalable inference and training of context-rich syntactic translation models. In *ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 961–968, Morristown, NJ, USA. Association for Computational Linguistics.

Qin Gao and Stephan Vogel. 2008. Parallel implementations of word alignment tool. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, SETQA-NLP '08, pages 49–57, Stroudsburg, PA, USA. Association for Computational Linguistics.

Hieu Hoang, Philipp Koehn, and Adam Lopez. 2009. A unified framework for phrase-based, hierarchical, and syntax-based statistical machine translation. In *In Proceedings of IWSLT, December 2009*.

Mark Hopkins and Greg Langmead. 2010. SCFG decoding without binarization. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 646–655, Cambridge, MA, October. Association for Computational Linguistics.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 48–54, Morristown, NJ, USA. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Morristown, NJ, USA. Association for Computational Linguistics.

Philipp Koehn, Barry Haddow, Philip Williams, and Hieu Hoang. 2010. More linguistic annotation for statistical machine translation. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 115–120, Uppsala, Sweden, July. Association for Computational Linguistics.

Daniel Marcu, Wei Wang, Abdessamad Echihabi, and Kevin Knight. 2006. SPMT: statistical machine translation with syntactified target language phrases. In *EMNLP '06: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 44–52, Morristown, NJ, USA. Association for Computational Linguistics.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Comput. Linguist.*, 29(1):19–51, March.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 160–167, Morristown, NJ, USA. Association for Computational Linguistics.

Andreas Stolcke. 2002. SRILM - an extensible language modeling toolkit. In *Intl. Conf. Spoken Language Processing, Denver, Colorado, September 2002*.

Hao Zhang, Liang Huang, Daniel Gildea, and Kevin Knight. 2006. Synchronous binarization for machine translation. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 256–263, Morristown, NJ, USA. Association for Computational Linguistics.

Andreas Zollmann and Ashish Venugopal. 2006. Syntax augmented machine translation via chart parsing. In *StatMT '06: Proceedings of the Workshop on Statistical Machine Translation*, pages 138–141, Morristown, NJ, USA. Association for Computational Linguistics.